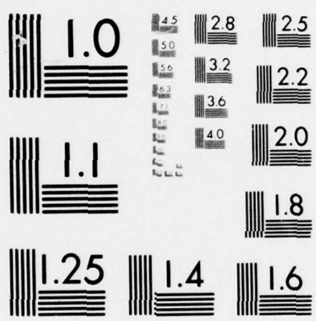
AD-A054 200 YALE UNIV NEW HAVEN CONN COWLES FOUNDATION FOR RESE--ETC F/G 15/5
SEALED BID AUCTIONS WITH NON-ADDITIVE BID FUNCTIONS,(U)
OCT 77 R ENGELBRECHT-WIGGANS N00014-77-C-0518
UNCLASSIFIED DISCUSSION PAPER-470 NL

1 OF 1
AD
A054200

END
DATE
FILMED
6-78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

1

COWLES FOUNDATION FOR RESEARCH IN ECONOMICS

AT YALE UNIVERSITY

Box 2125, Yale Station
New Haven, Connecticut 06520

14

COWLES FOUNDATION DISCUSSION PAPER [redacted] 470

Note: Cowles Foundation Discussion Papers are preliminary materials circulated to stimulate discussion and critical comment. Requests for single copies of a Paper will be filled by the Cowles Foundation within the limits of the supply. References in publications to Discussion Papers (other than mere acknowledgment by a writer that he has access to such unpublished material) should be cleared with the author to protect the tentative character of these papers.

6

SEALED BID AUCTIONS WITH NON-ADDITIVE BID FUNCTIONS

10 Richard Engelbrecht-Wiggans

12 21p.

11 31 Oct[redacted]77

15 N00014-77-C-0518

100 150

JOB

SEALED BID AUCTIONS WITH NON-ADDITIVE BID FUNCTIONS*

by

Richard Engelbrecht-Wiggans**

Abstract

A traditional sealed bid auction of a single item sells the item at the high bid price to a bidder with the highest bid. Such an auction may be used to auction several items; each bidder submits a bid on each item and each item is sold to a high bidder on that item. Implicit in this traditional scheme is the assumption that the bid for a set of items is the sum of the bids on the individual items: there are instances where this restriction appears unreasonable.

This paper considers a more general sealed bid auction in which bids are submitted on all possible subsets of the items. The items are partitioned among the bidders to maximize revenue, where each bidder pays what was bid on the set of items actually received. In general, the set partitioning problem is an extremely difficult integer programming problem, and there are two alternatives. The "greedy" and "sequential auction" heuristics are shown to result, at least for some examples, in very sub-optimal solutions. However, a class of slightly less general auction problems is presented for which optimal solutions may be calculated relatively easily; suggesting that some form of general sealed bid auctions may be appropriate in some situations.

---

*This work relates to Department of the Navy Contract N00014-77-C-0518 issued by the Office of Naval Research under Contract Authority NR 047-006. However, the content does not necessarily reflect the position or the policy of the Department of the Navy or the Government, and no official endorsement should be inferred.

**Parts of this research are based on portions of my doctoral dissertation [4], submitted to the Department of Operations Research at Cornell University. I wish to thank my thesis advisor, William F. Lucas, for his encouragement and Professor David C. Heath for stimulating my interest in auctions with non-additive bid functions.

The traditional sealed bid auction of a single item sells the item to a bidder with the highest bid. Such an auction may be used to auction several items; each bidder submits a sealed bid on each item and each item is assigned to a bidder with a high bid on that item. Implicit in this traditional sealed bid auction is the assumption that a bidder's bid for a set of items is the sum of the bids on individual items.

There are many instances where the restriction that bids be additive appears unreasonable. One example is a recent sale of offshore land for oil and gas exploration and development. There is considerable uncertainty about how much oil and gas is under any site and thus bidders are subject to considerable risk.

It seems reasonable to expect smaller oil companies to be more concerned about exposing themselves to a certain level of risk than a larger oil company, and that smaller companies must therefore somehow "hedge" their bids. Indeed, a recent Congressional Study [3] found that "it is with respect to the low interest, unpromising tracts that the smaller oil companies did the best" and that "in contrast, with respect to the high interest tracts, small companies won out over the majors in only 5 of the 52 tracts." This suggests that smaller companies "hedged" by either bidding only on less promising (lower expected value) sites or by bidding relatively small amounts for the more promising sites.

Any small company not "hedging" their bids subjects itself to the possibility that it will be awarded "too many" sites. In the actual sale, Conoco (which is considered a small company) was awarded 10.5% of the total lease area; this is second only to the area awarded to Exxon. However, soon thereafter, Gulf bought an interest in seven of the tracts

awarded to Conoco. This suggests that Conoco may have been awarded "too many" sites, thus forcing the sale of some of the award. This example suggests that it is not so much that "sale to oil firms are 'rigged'" as one newspaper headline stated [9], but that the traditional sealed bid auction is not well suited for allocating certain items.

One alternative is a very general sealed bid auction scheme suggested independently by several individuals, including Heath [7] and Vickrey [14]. This scheme allows bidders to submit bids on each possible subset of the collection of items being auctioned. The auctioneer assigns items to bidders to to maximize the revenue resulting from the auction; each bidder pays the amount bid for the subset of items actually assigned to that bidder. This general sealed bid auction would allow small oil companies to "hedge" by bidding relatively low on large sets of sites even though they may bid competitively on smaller sets.

As appealing as the scheme sounds, there are several difficulties. The first is that bidders must specify a large number of bids. However, this problem may be reduced if bidders let their bids be a function of bids on very small sets; for example, a bidder might specify bids on large sets by using the sum of the bids on individual items and then correcting this value by some fraction which reflects the size and/or riskiness of the large set. Thus, this difficulty seems surmountable.

A more serious difficulty is that the auctioneer must solve a set partitioning (or, alternatively, a subset selection) problem which is known to be, in general, an extremely difficult mathematical problem. Although branch and bound algorithms have been studied [1], there is no known technique for solving large problems of this class in a reasonable number of calculations. For small problems (for example, five bidders and twenty

items) it may be considered feasible to solve the integer program using dynamic programming. For larger problems, such an approach is impossible since the number of calculations required increases exponentially with the number of items to be auctioned.

The mathematical problems are not necessarily insurmountable. Two alternatives exist. One might consider an approximate solution to the integer program by some relatively simple heuristic if the heuristic always results in a solution with a value relatively close (with respect to the particular set of items being auctioned) to an optimal answer. In such cases, the inefficiencies of the heuristic may be smaller than the "inefficiencies" of alternative auction schemes.

A second alternative is to use any structure which might be present in a given situation. For example, in an auction for treasury bonds, it may be reasonable to assume that bidders are concerned only about how many (rather than which specific) bonds they are awarded; the bids should be a function only of the size of the subset of bonds. Using such structure, it may be possible to calculate optimal assignments relatively simply.

The first section of this paper introduces the notation used later, and defines the problem and the generalized auction scheme. The scheme is illustrated by a simple example.

Two heuristics for solving the integer program are defined and considered in the second section. The "greedy" heuristic is a commonly applied heuristic for solving integer programs and results in optimal solutions to knapsack problems related to the coin changing problem [2], [8], [10], [11]. However, the greedy heuristic is quite poor at obtaining optimal solutions to set-partitioning problems of the type arising from the generalized auction [4], [5], [6], [12], [13]. Even if the bid

functions are of a somewhat restricted form, the greedy heuristic may do as poorly as $1/m$ of the optimal solution value (where m is the number of goods to be auctioned). Thus, the greedy heuristic is in general an unlikely candidate for obtaining "good" approximate solutions.

An alternative to the greedy heuristic, is the very similar "sequential auction." However, an example illustrates that this heuristic is too inefficient for serious consideration. This result has implications beyond the schemes usefulness in solving set partitioning problems; the result suggests that certain auctions which are currently conducted sequentially would result in a greater total revenue if conducted as generalized sealed bid auctions.

Although the heuristics considered are not in general satisfactory for solving the set partitioning problem arising from the generalized sealed bid auction, the third section gives some more encouraging results for cases in which the bids functions have certain special structures. If each bidders bid function is a function only of the number (rather than which specific) items, then an optimal assignment may be calculated quite simply using dynamic programming. If in addition, the bid functions are such that each additional item has a decreasing marginal value to each bidder, then the greedy heuristic results in an optimal solution. Thus, there are relatively simple solution techniques for solving the set partitioning problem arising from auctions of homogeneous sets of items; possibly including treasury bonds, grain futures, junked cars, and production capacity of a machine shop. The section concludes by mentioning a slightly more general case of bid functions where dynamic programming may be used to solve problems with some divisible items.

1. Problem Statement

Let a set $M = \{1, 2, \ldots, m\}$ of m items be auctioned among n bidders by the following scheme. Each bidder i submits a sealed bid $v_i(s_i)$ for each subset s_i of M. The auctioneer then assigns (or "sells") the items according to a partition $\underline{s} = (s_1, s_2, \ldots, s_n)$ of M which maximizes $\Sigma_{i=1}^{i=n} v_i(s_i)$ over all possible partitions. Finally, each bidder i pays the corresponding $v_i(s_i)$ for receiving the goods s_i.

Numerous minor modifications to this scheme are possible. An arbitrarily negative bid $v_i(s_i)$ might be used to indicate that player i does not wish to bid on the subset s_i; or the auctioneer may specify reservation prices (below which an item will not be sold) by participating as a bidder and submitting appropriate bids. The results will be independent of many possible such minor modifications. It should be noted that this paper completely ignores the question of how the bidders decide what bids to submit; the paper assumes that the bids exist and studies the resulting set partitioning problem to help determine for what classes of real problems the generalized auction scheme is practicable.

The following example illustrates the auction scheme.

Example 1. Let there be two bidders for three items, and let

$s =$	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$v_1(s) =$	0	10	11	12	17	18	22	28
$v_2(s) =$	0	9	12	13	18	19	20	28 .

The unique revenue maximizing partition in example 1 is $(\{2,3\},\{1\})$; selling items 2 and 3 to the first bidder for 22 units and selling item

1 to the second bidder for 9 units results in a total revenue of 31. Although the revenue maximizing assignment is unique in this example, in general there may be several revenue maximizing assignments; this paper makes no attempt to decide which non-unique assignment should be used.

2. The "Greedy" and "Sequential Auction" Heuristics

A heuristic often considered in solving integer programs is the "greedy" heuristic. In the context of the problem being discussed, the greedy heuristic assigns items one by one until all items have been assigned. The heuristic first assigns the single item j to a player i such that $v_i(j)$ is maximized over all possible pairs of i and j. The heuristic next assigns the item which has the maximum marginal value, where the marginal value of item j to player i when player i has already been assigned the items in s_i is $v_i(s_i \cup j) - v_i(s_i)$. This last step is repeated until all items have been assigned; the step is repeated a total of m times.

The greedy heuristic may result in a suboptimal solution. For example, apply the greedy heuristic to example 1. First, item 3 is assigned to the second bidder. Next, since $v_1(2)$ exceeds $v_1(1)$, $v_2(1,3) - v_3(3)$, and $v_2(2,3) - v_2(3)$, the heuristic assigns item 2 to the first bidder. Finally, since $v_1(1,2) - v_1(2)$ is equal to $v_2(1,3) - v_2(3)$, item 1 may be assigned to either bidder. Either assignment results in a total revenue of 30 units; this is less than the optimal 31 units.

In general, the greedy heuristic may do much worse. If the $v_1(1,2,3)$ of example 1 is increased to 1,000,000 while all the other bids are as

before, then the optimal assignment is to assign all three items to the first bidder, with a resulting revenue of 1,000,000. The greedy solutions remain uneffected by this modification of the problem, but the ratio of the greedy solution value to the optimal is now only 3/100,000. Clearly, this can be carried to further extremes.

The greedy heuristic does quite poorly even when applied to a class of problems restricted to exclude the modification suggested above.

Lemma 1. Consider the class of set partitioning problems arising from auctions where the bid functions v_i satisfy the following three conditions:

1. $v_i(\emptyset) = 0$ for all i ;
2. $v_i(S) \leq v_i(T)$ for all i whenever $S \subseteq T$; and
3. $v_i(S) \leq \Sigma_{j \in S} v_i(j)$ for all i and for all S .

When applied to such problems, the greedy heuristic results in a solution with a value of at least $1/m$ of the optimal value and there are examples which give greedy solution values arbitrarily close to this lower bound.

Proof. The proof appears in [5] and the details will not be repeated here. The idea of the proof, however, is relatively simple. The lower bound is established by using the fact that the first item assigned by the greedy heuristic must have a value of at least $1/m$ of the optimal solution value, and noting that because of condition 2, the greedy solution value must be at least the value of the first item. The "tightness" of the bound is established by considering the following example.

<u>Example 2.</u> Let $n \geq 2$, $m \geq 2$, $0 < d < 1$, and

$v_1(1) = 1+d$, $v_1(s_1) = d - 1 + |s_1|$ if $|s_1| > 1$ and $1 \in s_i$,

and $v_1(s_1) = |s_1|$ if $1 \notin s_1$ (where $|S|$ = the number of elements in S);

$v_2(s_2) = d|s_2|$ if $1 \notin s_2$, and $v_2(s_2) = 1 - d + d|s_2|$ if $1 \in s_2$; and

$v_i(s_i) = 0$ for all s_i whenever $i > 2$.

The revenue maximizing assignment for this example is $s_1 = \{2, 3, \ldots, m\}$ and $s_2 = \{1\}$ with a value of m . The greedy solution is $s_1 = \{1\}$ and $s_2 = \{2, 3, \ldots, m\}$ with a value of $1 + md$. Thus the greedy solution value is arbitrarily close to $1/m$ of the optimal value. Notice that in addition to satisfying the three conditions of the lemma, the bid functions satisfy a more restrictive version of the third condition; namely $v_i(S) + v_i(T) \leq v_i(S \cup T)$ for all S and T and for all i . Thus, the bound in the lemma applies even to a more restricted class of set partitioning problems.

The poorness of the greedy solution in example 2 is due to the heuristic making a "bad" choice in what item to assign first. An alternative heuristic would be similar to the greedy heuristic but would not use the bid functions to determine the order in which items are assigned. In the "sequential auction" there is a predetermined order in which the items are assigned. This corresponds to a sequence of traditional sealed bid auctions for single items; at each stage a single item is sold to a highest bidder. Presumably, the bids at each stage reflect which (if

any) items a bidder has been assigned in previous stages. The auctions may actually be conducted sequentially in real time, or the bidders might submit their bid functions v_i and the auctioneer pretends that the auction was conducted sequentially.

For example, reconsider example 1 and assume the items are auctioned in numerical order. In the first auction, item 1 is sold to the first bidder for 10 units. In the second auction, the second bidder's bid of $v_2(2) = 10$ outbids the first bidder's bid of $v_1(1,2) - v_1(1) = 7$; item 2 is sold to the second bidder. Finally, since $v_1(1,3) - v_1(1) = 8$ $= v_2(2,3) - v_2(2)$, the third auction sells item 3 to either of the bidders.

In the above example, the total resulting revenue from either of the two sequential auction assignments is 30 units; the optimal value is 31. Thus, the sequential auction need not result in an optimal assignment. Indeed, it is easy to verify that regardless of the order in which items are auctioned, the sequential auction will always sell the first item to the "wrong" bidder in example 1; thus there is no order which results in an optimal solution.

The following example is one in which no possible order of auctioning items sequentially results value exceeding $R(n/m + 1/n)$, where R is the optimal solution value.

Example 3. Let $n > 1$, $m > 1$, and partition M into n sets $S_1, S_2, \ldots, S_n$ in any manner which equalizes the cardinalities of the S_i as much as possible. Thus, if $\text{Int}(x)$ denotes the integer part of x , $\text{Int}(m/n) \leq |S_i| < \text{Int}(m/n) + 1$ for all i . Let $0 < e < 1/m$, and consider the following bid functions.

$$v_i(s_i) = 1 + |(M\setminus S_i) \cap s_i| \quad \text{if} \quad S_i \cap s_i = \emptyset\,, \quad \text{and}$$

$$v_i(s_i) = e|(M\setminus S_i) \cap s_i| + |S_i \cap s_i| \quad \text{if} \quad S_i \cap s_i \neq \emptyset\,.$$

The sets S_i may be interpreted as "bidder i's " items; bidder i is willing to pay one unit for each item in S_i . The bidder is willing to pay $1+e$ units for any single item not in S_i , but additional items not in S_i are worth only e units additional. It may be verified that the revenue maximizing assignment awards each bidder i the corresponding set S_i ; the resulting revenue is m units.

Regardless of the order in which the items in example 3 are sold in a sequential auction, the first $n-1$ items will be sold to the "wrong" bidders; more precisely, for the first $n-1$ items sold, if item j is in S_i , item j will not be sold to bidder i . Notice that after the first $n-1$ items have been sold, the resulting marginal bids are such that as soon as some player i is sold an item from the corresponding S_i , then all the items auctioned subsequently will be sold to this same bidder i .

Thus, the best possible resulting assignment $\underline{s}$ has S_i a subset of s_i for some one bidder i , and $S_i \cap s_i$ empty for all other i . This implies that the maximum revenue is obtained when the n^{th} item to be auctioned is sold to the "correct" bidder (and even this possibility occurs only for more than two bidders) and all the remaining items are sold to this same bidder. The maximum revenue is $(n-1)(1+e)$ for the first $n-1$ items auctioned and $|S_i| + e(m-n+1-|S_i|)$ for selling the remaining items to a single bidder. Since, by assumption, $|S_i| < m/n+1$, the total revenue is less than $n+m/n+em$; and thus, by the assumption

on e, is less than $n + m/n = m(n/m + 1/n)$ (recall that m is the optimal solution value). This proves the following lemma.

Lemma 2. A sequential auction may result in less than $n/m + 1/n$ of the optimal solution value regardless of the order in which the items are auctioned.

If there are many more items than bidders, then this ratio tends to $1/n$; a rather small fraction of the optimal. For m not much larger than n, it is possible to construct examples in which $n^* < n$ of the bidders have bids as in example 3 and the remaining $n - n^*$ bidders are dummies (bidding zero on every set). The resulting ratio is then $n^*/m + 1/n^*$. Thus, a corollary of lemma 2 is the following.

Corollary 1. A sequential auction may result in less than $\text{minimum}_{1 \leq n^* \leq n,\ n^* \text{ integer}}(n^*/m + 1/n^*)$ of the optimal solution value regardless of the order in which items are auctioned.

For more than four items and at least two bidders, this ratio is less than one. It is substantially less than one for more than ten items. Thus, in general, it can not be expected that even the best possible order of auctioning items in a sequential auction is close to optimal. This not only limits the sequential auctions usefulness as a heuristic for solving the set partitioning problem, but also suggests that there are actual situations where sequential auctions are used but should be replaced with a more general auction scheme.

Finally, it should be noted that the revenue resulting from a sequential auction may be very sensitive to the actual bids. If, in example 3, $v_i(s_i)$ is set equal to $1 + e|\{(M \setminus S_i) \cap s_i\}| - 2e$, then the

sequential auction will always (regardless of the order in which items are auctioned) produce an optimal solution. An arbitrarily small change of $2e$ in the bids changes the revenue resulting from the (best possible) sequential auction by a factor of $m/n + 1/n$. Thus even in some situations in which bidders' true values are such that the sequential auction is efficient, any small variation from these true values (either from uncertainty in estimating the value or from not bidding actual values) may result in a substantially smaller revenue.

The above discussion illustrates some of the difficulties in solving the set partitioning problem arising from the general sealed bid auction scheme. Basically, either the problem is small enough to be solved directly using dynamic programming, or the problem is difficult to solve and even two common heuristics are not satisfactory. In particular, sequential auction schemes may be inappropriate.

3. Solutions for Structured Bid Functions

The previous section hints that the general sealed bid auction scheme may be infeasible in actual situations because the associated set partitioning problem is too difficult to solve. However, for any particular situation there may be sufficient structure to the bid functions such that there are solution procedures to solve the problem in a reasonable amount of time. This section presents several examples of class of bid functions whose structure result in set partitioning problems which are relatively easy to solve. Even though some of the examples may be considered applicable in themselves, the main point of this section is that there are cases of the general sealed bid auction which are practical, suggesting that there may be many actual problems where general auctions

are applicable.

There are many examples where bid functions depend only on the size of the set being bid on; this may happen whenever the collection of items to be auctioned consists of many similar items. Possible examples include stock tenders, treasury bonds, wheat futures, junked cars, corporate resources, hours of machine shop capacity, and temporary help in secretary pools. The calculation of exact solutions for bid functions depending only on the number of items is relatively simple using the dynamic programming algorithm presented below.

A slightly more general case than the above is when there are a number of different types of items. In particular, let there be k types of items and m_j items of type j for $j = 1, 2, \ldots, k$. If M_j denotes the set of m_j items of type j, then each bidder must specify a bid function from the points in $M_1 \times M_2 \times \ldots \times M_k$ to the real numbers. Notice that the original general sealed bid auction problem is the case when there is only one item of each of m types; the case when all items are of a single type was mentioned above. Thus, this model includes a number of special cases.

The following dynamic programming algorithm may be used to calculate exact solutions for problems where the bids are functions from $M_1 \times M_2 \times \ldots \times M_k$ to the real numbers.

<u>Algorithm.</u> 1. Let $i = 0$ and let $V_1(\underline{x}) \equiv V_1(x_1, x_2, \ldots, x_k)$ $= v_1(x_1, x_2, \ldots, x_k)$ for all $\underline{x}$ in $M_1 \times M_2 \times \ldots \times M_k$;

2. Increase i by one;

3. For each $\underline{x} \in M_1 \times M_2 \times \ldots \times M_k$, let $V_i(\underline{x}) = \max_{0 \leq \underline{y} \leq \underline{x}, \underline{y} \in M_1 \times M_2 \times \ldots \times M_k} (v_i(\underline{y}) + V_{i-1}(\underline{x}-\underline{y}))$ and let $\underline{T}_i(\underline{x})$ be a $\underline{y}$ which maximizes $V_i(\underline{x})$;

4. Repeat steps 2 and 3 until i is equal to n ;

5. Construct a revenue maximizing assignment by letting $\underline{t}_n \equiv (t_{n,1}, t_{n,2}, \ldots, t_{n,k}) = \underline{T}_n(\underline{m})$ where $\underline{m} \equiv (m_1, m_2, \ldots, m_k)$. For $i = n-1, n-2, \ldots, 2$ let $\underline{t}_i = \underline{T}_i(\underline{m} - \Sigma_{j=i+1}^{j=n}\underline{t}_j)$; and finally $\underline{t}_1 = \underline{m} - \Sigma_{j=2}^{j=n}\underline{t}_j$. Any assignment which assigns $t_{i,j}$ items of type j to bidder i for all pairs of i and j is a revenue maximizing assignment.

It may be verified that each iteration of step 3 requires a total of $\Pi_{j=1}^{j=k}(m_j + 1)(m_j + 2)/2$ elementary function evaluations and comparisons if the maxima are calculated by enumerating all possibilities. By storing the values of the computational variables appropriately, step 1 requires no computations; steps 2 and 4 each require one addition or comparison for each of the iterations. Step 5 requires calculating k coordinates for each of n vectors, or a total of on the order $n \cdot k$ operations. It is clear that essentially all the computations required occur in step 3, and thus the total of $(n-1)\Pi_{j=1}^{j=k}(m_j + 1)(m_j + 2)/2$ computations is a reasonable measure of the difficulty of solving a problem using this algorithm.

It may also be verified that a total of $(n-1)\Pi_{j=1}^{j=k}(m_j + 1)$ vectors $\underline{T}_j(\underline{x})$, each with k coordinates, must be recorded and n vectors $\underline{t}_j$, each with k coordinates, must also be recorded. Thus, aside from any storage required to record the bid functions, the algorithm requires storage for approximately $(n-1)(k+1)\Pi_{j=1}^{j=k}(m_j + 1)$ numbers.

Using these measures of the computational and storage requirements, it is possible to obtain some idea what size problems may be solved on

a computer. For example, a problem with ten bidders and thirty items of each of three different types requires approximately $9\cdot(31\cdot32/2)^3$ or about 10^9 elementary arithmetic operations and approximately $9\cdot4\cdot31^3$ or about two million numbers to be stored. A problem of this size may be solved in a few hours on a large computer.

Two special cases of the algorithm are of special interest. The first is where there is only one item of each of m types; there are m different items. This corresponds to the completely general sealed bid auction described in the first section. In this case, the algorithm requires approximately $(n-1)3^m$ computations and $2n2^m$ storage locations. While problems of assigning fifteen items among five bidders may be solved in a reasonable amount of computer time and storage, the difficulty grows exponentially with the size of the auction problem. Problems of assigning more than twenty items among more than ten bidders are essentially impossible to solve using this algorithm. It is this difficulty that prompted the consideration of the heuristics in the second section and the consideration of slightly less general auctions in this section.

One less than completely general class of problems is when there are a moderate number of items, but only a few different types; the complexity of the set-partitioning problem is very dependent on the number of different types. As previously mentioned, problems with $n = 10$, $k = 3$, and $m_1 = m_2 = m_3 = 30$ can be solved on a computer. The extreme case, a second special case, is when there is only one type of item.

When there are m items of a single type, the bid functions depend only on the number of items in the set being bid on. In this case, the algorithm requires approximately $(n-1)(m+1)(m+2)/2$ computations and

$2n(m+1)$ storage locations. Thus even relatively large problems of assigning 3000 items among 30 bidders can be solved on a computer. Unlike the previous case, the amount of computation and storage does not grow exponentially with the size of the problem; if all items are of one type, quite large problems may be easily solved.

If there is only one type of item and the bid functions are restricted yet a little further, then the greedy heuristic will give optimal solutions. In particular, a bid function is defined to have decreasing marginal values if it depends only on the number of items in a set and $v_i(k+1) - v_i(k) \leq v_i(k) - v_i(k-1)$ for $k = 1, 2, \ldots, m-1$.

A corollary of the following lemma is that the greedy heuristic yields optimal solutions when all bid functions have decreasing marginal values. In such cases the number of computations required in calculating an optimal solution is on the order of $n \cdot m$; problems with over a million items may be easily solved.

<u>Lemma 4</u>. If the bid functions depend only on the number of items in the set being bid on, then a solution assigning k_i items to bidder i (for $i = 1, 2, \ldots, n$) is optimal (or, equivalently, revenue maximizing) if there exists a real number v^* such that the following conditions are satisfied.

1. $v^* \leq (v_i(k_i) - v_i(j))/(k_i - j)$ for $j = 0, 1, \ldots, k_i - 1$ and $i = 1, 2, \ldots, n$; and
2. $v^* \geq (v_i(j) - v_i(k_i))/(j - k_i)$ for all $j = k_i + 1, \ldots, m$ and $i = 1, 2, \ldots, n$.

Proof. Assume that there is an assignment satisfying the above conditions. Now consider any alternative assignment which assigns k_i' to bidder i (for $i = 1, 2, \ldots, n$), and assume the bidders are numbered such that $k_i < k_i'$ for $i = 1, 2, \ldots, n_1$; $k_i > k_i'$ for $i = n_1 + 1, n_1 + 2, \ldots, n_1 + n_2$; and $k_i = k_i'$ for the remaining i .

For the first n_1 bidders, the first condition of the lemma implies that $v_i(k_i) \geq v_i(k_i') + (k_i - k_i')v^*$. Likewise, the second condition implies $v_i(k_i) \geq v_i(k_i') - (k_i' - k_i)$ for bidders $n_1 + 1$ through $n_1 + n_2$. However, since $\Sigma_{i=1}^{i=n} k_i = \Sigma_{i=1}^{i=n} k_i'$, it follows that $\Sigma_{i=1}^{i=n}(k_i - k_i')$ $= -\Sigma_{i=n_1+1}^{i=n_1+n_2}(k_i' - k_i)$; summing the above inequalities for $v_i(k_i)$ and simplifying gives that $\Sigma_{i=1}^{i=n} v_i(k_i) \geq \Sigma_{i=1}^{i=n} v_i(k_i')$. Thus, assigning k_i items to bidder i (for $i = 1, 2, \ldots, m$) must be revenue maximizing.

The following corollary states conditions under which the greedy heuristic gives optimal solutions.

Corollary 2. If all bidders have bid functions with decreasing marginal values, then the greedy heuristic yields revenue maximizing assignments of the items.

Proof. Let j be the bidder receiving the last item to be assigned, and then let $v^* = v_j(k_j) - v_j(k_j - 1)$ where k is the total number of items assigned to bidder j . Clearly, this v^* satisfies the conditions of lemma 3 for bid functions with decreasing marginal values when the items are assigned using the greedy heuristic.

Unfortunately, there does not appear to be any similar restriction for problems with more than one type of item so that the greedy heuristic

results in optimal solutions. This conclusion results from a quick examination of example 1 in which there is one item of each of three types. Although it is not certain how the concept of "decreasing marginal values" should be extended to items of different types, example 1 makes it clear that it is not sufficient to assume subadditivity in order to ensure the optimality of greedy solutions.

Finally, it should be noted that although all the above assumes that the m items to be auctioned are indivisible items, a similar generalized sealed bid auction scheme may be defined for cases where some of the items are finely divisible. If items one through k are divisible, then the bid functions must be functions from $[0,1]^k \times \{0,1\}^{m-k}$ to the real numbers.

A "continuous greedy" heuristic may be defined, and the recursion equation (step 3 of the dynamic programming algorithm) must be generalized to real numbers rather than just integers. Conceptually, this more general view does not cause any difficulties. From a computational point of view, however, it is quite possible that the functions involved are not sufficiently "nice" for it to be possible for a computer to solve step 3 of the recursion.

Alternatively, if all the items are indivisible, and the bid functions are sufficiently "nice," it may be able to solve the recursion equation without explicitly enumerating all possible integer values. Such a situation would further simplify the calculation of optimal solutions.

Hopefully, further work will identify additional classes of bid functions corresponding to actual auctions for which satisfactory assignments may be easily calculated. Such results would provide additional support for the consideration of generalized sealed bid auctions.

REFERENCES

[1] D. E. Boyce; A. Farhi; and R. Weischedel. "Optimal Subset Selection," Lecture Notes in Economics and Mathematical Systems (No. 103), Springer Verlag, New York, 1975.

[2] L. Chang and J. F. Korsh. "Canonical Coin Changing and Greedy Solutions," J. Assoc. Comput. Machinery, No. 23 (1976), pp. 418-422.

[3] Congressional Study: An analysis of the economic impact of the current OCS bidding system, prepared for Representative William Hughs (D-NJ), 1976.

[4] R. Engelbrecht-Wiggans. "On the Fair and Efficient Allocation of Indivisible Commodities," School of Operation Research and Industrial Engineering Technical Report No. 356, Cornell University, 1977.

[5] R. Engelbrecht-Wiggans. "The Greedy Heuristic Applied to a Class of Set Partitioning and Subset Selection Problems," Cowles Foundation Discussion Paper No. 469, 1977.

[6] M. L. Fisher; G. L. Nemhauser; and L. A. Wolsey. "An Analysis of Approximations for Maximizing Submodular Set Functions II," CORE Discussion Paper No. 7629, December 1976.

[7] D. C. Heath, private communication.

[8] T. C. Hu and M. L. Lenard. "Optimality of a Heuristic Algorithm for a Class of Knapsack Problems," Opns. Res., No. 24 (1976), pp. 193-196.

[9] Ithaca Journal. "Sales to oil firms are 'rigged,' report says," December 10, 1976; Ithaca, New York.

[10] S. C. Johnson and B. W. Kernighan. "Making Change with a Minimum Number of Coins," Internal Technical Memorandum, Bell Telephone Laboratories, Murray Hill, New Jersey, 1971.

[11] M. Magazine; G. L. Nemhauser; and L. E. Trotter, Jr. "When the Greedy Solution Solves a Class of Knapsack Problems," Opns. Res., No. 23, (1975), pp. 203-217.

[12] G. L. Nemhauser; M. L. Fisher; and L. A. Wolsey. "An Analysis of Approximations for Maximizing Submodular Set Functions," CORE Discussion Paper No. 7618, July 1976.

[13] G. L. Nemhauser; L. A. Wolsey; and M. L. Fisher. "Best Algorithms for Approximating the Maximum of a Submodular Set Function," CORE Discussion Paper No. 7636, December 1976.

[14] W. Vickrey. "Auctions, Markets, and Optimal Allocation," in Bidding and Auctions for Procurement and Allocation, edited by Yakov Amihud, New York University Press, New York, 1976, pp. 13-20.